UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/631,174	08/02/2000	Lawrence D. K. B. Dwyer	10001219-1	7798

7590 02/08/2006

Hewlett-Packard Company
Intellectual Property Administration
P O Box 272400
Fort Collins, CO 80527-2400

EXAMINER	
LI, AIMEE J	
ART UNIT	**PAPER NUMBER**
2183	

DATE MAILED: 02/08/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/631,174	DWYER ET AL.	
	Examiner	Art Unit	
	Aimee J. Li	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 November 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4-8,10-16,18-27 and 30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4-8,10-16,18-27 and 30* is/are rejected.

7)☒ Claim(s) *7,11 and 19* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-2, 4-8, 10-16, 18-27, and 30 have been considered. Claims 3, 9, 17, 28, and 29 have been cancelled as per Applicant's request. Claims 1, 7, 10, 11, 12, 16, 18-22, and 30 have been amended as per Applicant's request.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: RCE as received on 25 November 2005 and Amendment as received on 25 November 2005.

Claim Objections

3. Claim 7 is objected to because of the following informalities: Claim 7 recites "preload, into said cache memory, said at least on e data value..." please correct it to read --preload, into said cache memory, said at least [[on e]]one data value...--. Deletions are denoted by strikethroughs or double brackets and additions are denoted by underlines. Appropriate correction is required.

4. Claim 11 is objected to because of the following informalities: Claim 11 recites "The system of claim 9..." please correct it to read -- The system of claim [[9]]7...-- since claim 9 has been cancelled. Deletions are denoted by strikethroughs or double brackets and additions are denoted by underlines. Appropriate correction is required.

5. Claim 19 is objected to because of the following informalities: Claim 19 recites "The system of claim 17..." please correct it to read -- The system of claim [[17]]16...-- since claim 17 has been cancelled. Deletions are denoted by strikethroughs or double brackets and additions are denoted by underlines. Appropriate correction is required.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1-2, 4-8, 10-16, 18-27, and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lass, U.S. Patent Number 6,092,153 (herein referred to as Lass), in view of Tremblay et al., U.S. Patent Number 6,205,543 (herein referred to as Tremblay).

8. Referring to claim 1, Tremblay has taught a computer system for efficiently executing instructions of computer programs, comprising:

a. Said processing circuitry configured to stop executing said one program during a first context switch in response to a first context switch command and to resume executing said one program during a second context switch in response to a second context switch command (Lass column 1, lines 62-67 and column 6, lines 9-28);

b. Cache memory (Lass column 1, lines 62-67; column 4, lines 54-59; column 5, lines 29-30; and column 6, lines 9-28); and

c. Computer memory having a plurality of addresses (Lass column 1, lines 62-67; column 4, lines 54-59; column 5, lines 29-30; and column 6, lines 9-28).

9. Lass has not taught

a. Processing circuitry having a pipeline, said pipeline configured to execute instructions from one of a plurality of programs, and

b. Memory control circuitry coupled to said processing circuitry, said memory control circuitry configured to store, in response to said first context switch command, in computer memory, data written by said pipeline during execution of said one program and to store an indicator indicative of whether said data was accessed by the processing circuitry during a particular time period prior to said first context switch in executing said instructions from said one program, said memory control circuitry, in response to said second context switch command, configured to identify one of said addresses of said computer memory that is storing said indicator corresponding to said data previously written by said pipeline during execution of an instruction of said one computer program prior to said first context switch, said memory control circuitry further configured to make a determination, based on said indicator, whether said data was accessed by the processing circuitry in the particular time period prior to the first context switch for determining whether to preload said data into said cache memory in response to said second context switch, to retrieve said data, based upon said determination, from said computer memory in response to said second context switch command, and to store said retrieved data in said cache memory based upon said indicator.

10. However Lass has taught in column 4, lines 54-59 that a partial save and restore of cache information on a context switch but provided no details. Tremblay has taught the details to partially saving and restoring data with

a. Processing circuitry having a pipeline, said pipeline configured to execute instructions from one of a plurality of programs (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 5; and Figure 6), and

b. Memory control circuitry coupled to said processing circuitry, said memory control circuitry configured to store, in response to said first context switch command, in computer memory, data written by said pipeline during execution of said one program and to store an indicator indicative of whether said data was accessed by the processing circuitry during a particular time period prior to said first context switch in executing said instructions from said one program (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 5; and Figure 6), said memory control circuitry, in response to said second context switch command, configured to identify one of said addresses of said computer memory that is storing said indicator corresponding to said data previously written by said pipeline during execution of an instruction of said one computer program prior to said first context switch, said memory control circuitry further configured to make a determination, based on said indicator, whether said data was accessed by the processing circuitry in the particular time period prior to the first context switch for determining whether to preload said data into said cache memory in response to said second context switch, to retrieve said data,

based upon said determination, from said computer memory in response to said second context switch command, and to store said retrieved data in said cache memory based upon said indicator (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 5; and Figure 6).

11. A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Lass and Tremblay, that saving and restoring portions of data memory creates a more refined system (Lass column 4, lines 57-59) and improves the efficiency of context switching in a multiprogrammed processor (Tremblay column 2, lines 4-6). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the saving and restoring portions of data memory based upon marker bits, e.g. dirty bits, to have a more refined system with improved context switching efficiency.

12. Referring to claim 2, Lass in view of Tremblay has taught wherein said processing circuitry is further configured to execute instructions of another of said computer programs in response to said first context switch command (Lass column 1, lines 62-67; column 4, lines 54-59; column 5, lines 29-30; and column 6, lines 9-28).

13. Referring to claim 4, Lass in view of Tremblay has taught

a. Said cache memory (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28); and

b. Wherein said memory control circuitry is configured to maintain a plurality of mappings, each of said mappings respectively correlating data stored in said data

memory with one of said memory addresses of said computer memory, said memory control circuitry further configured to maintain a bit of information that is associated with one of said mappings, said memory control circuitry configured to assert said bit when a data value correlated with a computer memory address via said one mapping is utilized to execute an instruction of said one program, said memory control circuitry further configured to deassert said bit periodically (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6).

14. Referring to claim 5, Lass in view of Tremblay has taught wherein said memory control circuitry is further configured to determine, in response to said second context switch command and based on said bit, whether said data value was recently utilized by said processing circuitry to execute an instruction prior to said first context switch (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6).

15. Referring to claim 6, Lass in view of Tremblay has taught wherein said memory control circuitry is further configured to store said mappings and said bit to said computer memory in response to said first context switch command and to retrieve said mappings and said bit from said computer memory in response to said second context switch command (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28).

16. Referring to claim 7, Lass has taught a computer system for efficiently executing instructions of computer programs, comprising:

a. Said processing circuitry configured to stop executing said one program during a first context switch in response to a first context switch command and to resume executing said one program during a second context switch in response to a second context switch command (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28);

b. Cache memory (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28); and

c. Computer memory having a plurality of addresses (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28).

17. Lass has not taught

a. Processing circuitry having a pipeline, said pipeline configured to execute instructions nom one of a plurality of programs (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6),

b. Memory control circuitry coupled to said processing circuitry, said memory control circuitry configured to maintain a plurality of mappings, said mappings respectively correlating at least one data value previously written by said pipeline during execution of an instruction and stored in said data memory with said memory addresses of said computer memory, said memory control circuitry configured to store in said computer memory said mappings and information indicating whether said at least one data value was recently accessed and to make

a determination, based on said information, whether said information indicates that said at least one data value was recently accessed prior to the first context switch, said memory control circuitry further configured to preload, into said data memory, said at least one data value based on said determination if said information indicates that said at least one data value was recently accessed prior to said first context switch.

18. However Lass has taught in column 4, lines 54-59 that a refinement to the threading system is a partial save and restore of cache information on a context switch but provided no details. Tremblay has taught the details to partially saving and restoring data in data memory with

a. Processing circuitry having a pipeline, said pipeline configured to execute instructions nom one of a plurality of programs (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6),

b. Memory control circuitry coupled to said processing circuitry, said memory control circuitry configured to maintain a plurality of mappings, said mappings respectively correlating at least one data value previously written by said pipeline during execution of an instruction and stored in said data memory with said memory addresses of said computer memory (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and

Figure 6), said memory control circuitry configured to store in said computer memory said mappings and information indicating whether said at least one data value was recently accessed and to make a determination, based on said information, whether said information indicates that said at least one data value was recently accessed prior to the first context switch, said memory control circuitry further configured to preload, into said data memory, said at least one data value based on said determination if said information indicates that said at least one data value was recently accessed prior to said first context switch (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6).

19. A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Lass and Tremblay, that saving and restoring portions of data memory creates a more refined system (Lass column 4, lines 57-59) and improves the efficiency of context switching in a multiprogrammed processor (Tremblay column 2, lines 4-6). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the saving and restoring portions of data memory based upon marker bits, e.g. dirty bits, to have a more refined system with improved context switching efficiency.

20. Referring to claim 8, Lass in view of Tremblay has taught wherein said processing circuitry is further configured to execute instructions of another of said computer programs in

response to said first context switch command (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28).

21. Referring to claim 10, Lass in view of Tremblay has taught wherein said memory control circuitry is further configured to store said information in said computer memory in response to said first context switch command and to retrieve said information and said mappings in response to said second context switch command (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28).

22. Referring to claim 11, Lass in view of Tremblay has taught wherein said information has a plurality of bits respectively associated with said mappings, wherein said memory control circuitry, for each data value accessed by said memory control circuitry, is configured to assert the bit associated with the mapping that correlates said each data value with one of said computer memory addresses, and wherein said memory control circuitry is configured to periodically deassert each of said plurality of bits (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6).

23. Referring to claim 12, Lass has taught a method for efficiently executing instructions of computer programs, comprising the steps of:

a. Executing a plurality of computer programs in an interleaved fashion (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28);

b. Switching which of said computer programs is being executed in said executing step (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28);

c. Identifying said address in response to said switching step (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28); and

d. Storing said retrieved data value in cache memory (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28).

24. Lass has not taught

a. Storing, prior to said switching step, at an address in computer memory a data value previously written by a pipeline to a data line in execution of an instruction corresponding to one of said computer programs in said executing step and an indicator indicating if the data line was recently accessed;

b. Determining, based on said indicator, whether the data value was recently accessed prior to said switching step; and

c. Retrieving, based on said determining step, said data value from said address based on said identifying step and in response to said switching step if said indicator indicates that the data was recently accessed.

25. However Lass has taught in column 4, lines 54-59 that a refinement to the threading system is a partial save and restore of cache information on a context switch but provided no details. Tremblay has taught the details to partially saving and restoring data in data memory with

a. Storing, prior to said switching step, at an address in computer memory a data value previously written by a pipeline to a data line in execution of an instruction corresponding to one of said computer programs in said executing step and an indicator indicating if the data line was recently accessed (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6);

b. Determining, based on said indicator, whether the data value was recently accessed prior to said switching step (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6); and

c. Retrieving, based on said determining step, said data value from said address based on said identifying step and in response to said switching step if said indicator indicates that the data was recently accessed (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6).

26. A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Lass and Tremblay, that saving and restoring portions of data memory creates a more refined system (Lass column 4, lines 57-59) and improves the efficiency of context switching in a multiprogrammed processor (Tremblay column 2, lines 4-6). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the

invention was made to incorporate the saving and restoring portions of data memory based upon marker bits, e.g. dirty bits, to have a more refined system with improved context switching efficiency.

27. Referring to claims 13 and 15, Lass in view of Tremblay has taught wherein said executing step further includes the step of executing instructions of a computer program in response to said switching step, and wherein said method further comprises the steps of:

a. Determining that said address is storing a data value previously utilized in said executing step to execute an instruction of said computer program (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6); and

b. Performing said identifying step based on said determining step (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6).

28. Referring to claim 14, Lass in view of Tremblay has taught

a. Said cache memory (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28);

b. Correlating, respectively, data values stored in said data memory with addresses of said computer memory (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6);

c. Asserting a bit each time a data value correlated with said address identified in said identifying step is accessed in response to said executing step (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6); and

d. Periodically deasserting said bit (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6).

29. Referring to claim 16, Lass has taught a method for efficiently executing instructions of computer programs, comprising the steps of:

a. Executing instructions from a computer program (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28);

b. Halting said executing step during a first context switch in response to a first context switch command (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28);

c. Resuming said executing step during a second context switch in response to a second context switch command (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28);

d. Cache memory (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28); and

e. Storing said at least one retrieved data value in said cache memory (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28).

30. Lass has not taught

a. Maintaining a plurality of mappings;

b. Correlating, via said mappings, data values previously written by a pipeline during the executing step and stored in a data memory with memory addresses of computer memory outside of said data memory;

c. Storing said mappings in said computer memory in response to said first context switch command and information indicative of whether said data values were accessed during a particular time period prior to said first context switch;

d. Selecting, based on said information and for preloading into the data memory in response to the second context switch command, at least one data value from at least one of said addresses identified by said mappings; and

e. Retrieving, based on said mappings and said selecting step, said at least one data value in response to said second context switch command if said information indicates that said at least one data value was accessed during said particular time period.

31. However Lass has taught in column 4, lines 54-59 that a refinement to the threading system is a partial save and restore of cache information on a context switch but provided no details. Tremblay has taught the details to partially saving and restoring data in data memory with

a. Maintaining a plurality of mappings (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6);

b. Correlating, via said mappings, data values previously written by a pipeline during the executing step and stored in a data memory with memory addresses of computer memory outside of said data memory (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6);

c. Storing said mappings in said computer memory in response to said first context switch command and information indicative of whether said data values were accessed during a particular time period prior to said first context switch (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6);

d. Selecting, based on said information and for preloading into the data memory in response to the second context switch command, at least one data value from at least one of said addresses identified by said mappings (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6); and

e. Retrieving, based on said mappings and said selecting step, said at least one data value in response to said second context switch command if said information indicates that said at least one data value was accessed during said particular time period (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6).

32. A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Lass and Tremblay, that saving and restoring portions of data memory creates a more refined system (Lass column 4, lines 57-59) and improves the efficiency of context switching in a multiprogrammed processor (Tremblay column 2, lines 4-6). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the saving and restoring portions of data memory based upon marker bits, e.g. dirty bits, to have a more refined system with improved context switching efficiency.

33. Referring to claim 18, Lass in view of Tremblay has taught

a. Storing said information in said computer memory in response to said first context switch command (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28); and

b. Retrieving said information and said mappings in response to said second context switch command (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28).

34. Referring to claim 19, Lass in view of Tremblay has taught wherein said information has a plurality of bits respectively associated with said mappings, and wherein said method further comprises the steps of:

a. Asserting each of said bits associated respectively with each of said mappings that identifies a data value accessed in response to said executing step (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6); and

b. Periodically deasserting each of said bits (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6).

35. Referring to claim 20, Lass has taught a computer system for efficiently executing instructions of computer programs, comprising:

a. Computer memory (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28); and

b. A processing unit comprising cache memory (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28) and

c. Said processing unit continuing execution of said first process with the retrieved data when the processing unit context switches out the second process and context switches in the first process (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28).

36. Lass has not taught

a. Logic configured to store in said computer memory a value indicative of whether a portion of said cache memory was recently accessed by said processor and a mapping associated with said value, said mapping indicative of a location in said computer memory storing data previously requested or previously written by an instruction of a first process being executed by the processing unit when the processing unit context switches out the first process for processing of a second process, the logic further configured to select said data for preloading in said cache memory after said second context switch if said value indicates that the data was recently accessed, retrieve said data based on said value and store said data in said cache if said value indicates that the data was recently accessed.

37. However Lass has taught in column 4, lines 54-59 that a refinement to the threading system is a partial save and restore of cache information on a context switch but provided no details. Tremblay has taught the details to partially saving and restoring data in data memory with

a. Logic configured to store in said computer memory a value indicative of whether a portion of said cache memory was recently accessed by said processor and a mapping associated with said value (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6), said mapping indicative of a location in said computer memory storing data previously requested or previously written by an instruction of a first process being executed by the processing unit when the processing unit context switches

out the first process for processing of a second process (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6), the logic further configured to select said data for preloading in said cache memory after said second context switch if said value indicates that the data was recently accessed, retrieve said data based on said value and store said data in said cache if said value indicates that the data was recently accessed (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6).

38. A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Lass and Tremblay, that saving and restoring portions of data memory creates a more refined system (Lass column 4, lines 57-59) and improves the efficiency of context switching in a multiprogrammed processor (Tremblay column 2, lines 4-6). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the saving and restoring portions of data memory based upon marker bits, e.g. dirty bits, to have a more refined system with improved context switching efficiency.

39. Referring to claim 21, Lass has taught a method for efficiently executing instructions of computer programs, comprising the steps of using cache memory and context switches (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28). Lass has not taught

a. Storing a value indicative of whether data in data memory was recently accessed by a processing unit;

b. Storing in said memory a mapping corresponding to said value, said mapping indicative of a location in computer memory storing said data when the processing unit context switches out the first process for processing of a second process;

c. Determining whether said data was recently accessed prior to said processing unit switching out the first process;

d. Preloading, based on said determining, step, said data in said data for execution of said first process by said processing unit if said value indicates that the data was recently accessed in said determining step.

40. However Lass has taught in column 4, lines 54-59 that a refinement to the threading system is a partial save and restore of cache information on a context switch but provided no details. Tremblay has taught the details to partially saving and restoring data in data memory with

a. Storing a value indicative of whether data in data memory was recently accessed by a processing unit (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6);

b. Storing in said memory a mapping corresponding to said value, said mapping indicative of a location in computer memory storing said data when the processing unit context switches out the first process for processing of a second

process (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6);

c. Determining whether said data was recently accessed prior to said processing unit switching out the first process (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6);

d. Preloading, based on said determining, step, said data in said data for execution of said first process by said processing unit if said value indicates that the data was recently accessed in said determining step (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6).

41. A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Lass and Tremblay, that saving and restoring portions of data memory creates a more refined system (Lass column 4, lines 57-59) and improves the efficiency of context switching in a multiprogrammed processor (Tremblay column 2, lines 4-6). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the saving and restoring portions of data memory based upon marker bits, e.g. dirty bits, to have a more refined system with improved context switching efficiency.

42. Referring to claim 22, Lass in view of Tremblay has taught wherein said cache memory comprises a cache line (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28).

43. Referring to claim 23, Lass in view of Tremblay has taught wherein said value is indicative of whether said processing unit has accessed said cache line (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28) during a particular time period (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6).

44. Referring to claim 24, Lass in view of Tremblay has taught wherein said value indicative of said cache memory (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28) usage is defined by a flag, said logic configured to assert said flag when said first process uses said cache line (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6).

45. Referring to claim 25, Lass in view of Tremblay has taught wherein said cache memory comprises a cache line (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28).

46. Referring to claim 26, Lass in view of Tremblay has taught wherein said value is indicative of whether said processing unit has accessed said cache line (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28) during a particular time period (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21;

column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6).

47. Referring to claim 27, Lass in view of Tremblay has taught the step of asserting a flag when said first process uses said cache line (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28) (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6).

48. Referring to claim 30, Lass has taught a system, comprising:

a. Computer memory (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28); and

b. A processing unit comprising cache memory (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28), said cache memory comprising a plurality of cache lines (Lass column 1, lines 62-67; column 4, lines 54-59; column 4, lines 29-30; and column 6, lines 9-28).

49. Lass has not explicitly taught storing data written or read by a first process during execution by said processing unit, said processing unit further comprising logic configured to periodically assert one of a plurality of flags associated with each of said data lines when said flag's associated data line is accessed, said logic further configured to store in said computer memory said flags and data corresponding to the associated data lines accessed by said first process upon a first context switch, said logic further configured to select data for preloading into the data memory based upon the asserted flags in response to a based on said information. However Lass has taught in column 4, lines 54-59 that a refinement to the threading system is a

partial save and restore of cache information on a context switch but provided no details. Tremblay has taught the details to partially saving and restoring data in data memory with storing data written or read by a first process during execution by said processing unit, said processing unit further comprising logic configured to periodically assert one of a plurality of flags associated with each of said data lines when said flag's associated data line is accessed, said logic further configured to store in said computer memory said flags and data corresponding to the associated data lines accessed by said first process upon a first context switch, said logic further configured to select data for preloading into the data memory based upon the asserted flags in response to a based on said information (Tremblay Abstract; column 2, lines 10-31; column 2, line 55 to column 3, line 21; column 5, lines 1-19; column 6, lines 35-43; column 10, line 30 to column 12, line 2; Figure 2; Figure 5; and Figure 6). A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Lass and Tremblay, that saving and restoring portions of data memory creates a more refined system (Lass column 4, lines 57-59) and improves the efficiency of context switching in a multiprogrammed processor (Tremblay column 2, lines 4-6). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the saving and restoring portions of data memory based upon marker bits, e.g. dirty bits, to have a more refined system with improved context switching efficiency.

Response to Arguments

50. Applicant's arguments with respect to claims 1-2, 4-8, 10-16, 18-27, and 30 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

51. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J. Li whose telephone number is (571) 272-4169. The examiner can normally be reached on M-T 7:00am-4:30pm.

52. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

53. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AJL
Aimee J. Li
1 February 2006

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100